

(43) Date of publication of application: **17.03.95**

G06F 9/46
G06F 11/30

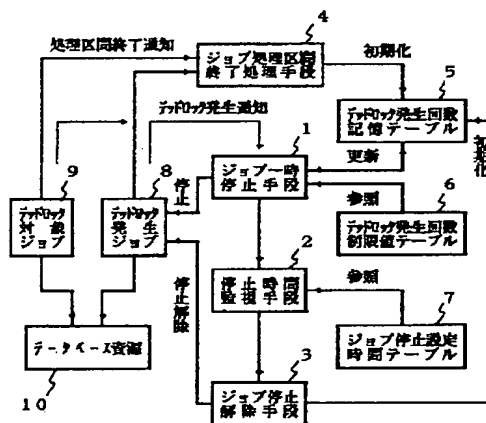
(71) Applicant: **WAKAYAMA NIPPON DENKI
SOFTWARE KK**

(72) Inventor: **NAKAGAWA NOBORU**

the deadlock occurrence frequency storage table 5 at the time of the end of each job processing section.

COPYRIGHT: (C)1995,JPO

CONSTITUTION: This system is provided with a deadlock occurrence frequency storage table 5 where the frequency in occurrence of deadlock in the processing section executed at present is integrated and held, a job temporary stop meant 1 which receives the report of the occurrence of deadlock to add one to the frequency in occurrence of the job and temporarily stops the job at the time of arrival at the limit value in a deadlock occurrence frequency limit value table 6, a stop time monitor means 2 which monitors the temporary stop time and issues the end report at the time of the elapse of a prescribed time set to a job stop set time table 7, a job stop release means 3 which receives the end report to release the temporary stop of the job and also initialize the frequency in occurrence of the job in the deadlock occurrence frequency storage table 5, and a job processing section termination processing means 4 which initializes the frequency in occurrence of the job in



JP 3005397

[0002]

[Prior Art]

In an information processing system such as a database system for performing parallel processing while at least two jobs are using the same resource exclusively, a deadlock phenomena that the two jobs request resources occupied mutually by the mutually mating jobs and they enter in a waiting state may occur. As a measure to avoid such a phenomena, such a process is employed that, when a deadlock occurs, executed processing in currently executing processing section is cancelled to a deadlock occurrence job which has generated the later resource request to cause a deadlock state and re-execution is performed from a starting point of the processing section. This deadlock can generally be avoided by this measure.

[0003]

As mentioned above, a deadlock has occurred at a time point within the processing section is avoided by re-executing the deadlock occurrence job from the starting point, and the deadlock object job which is a cause of the deadlock can continue processing following the processing at the time of the deadlock occurrence. However, when the processing section is long and complicated, another deadlock phenomena occurs again at another time in the same processing section, and when the own job is the deadlock occurrence job, processing can not be continued

so that the processing is returned back to the starting point of the processing section. Accordingly, when two jobs causes deadlocks alternatively in one processing section to repeat the deadlock, a phenomena (which is called a multi-occurrence deadlock) where both the jobs only repeat a starting processing (for example, database update processing) from the processing section starting point and an update canceling processing due to the deadlock alternatively and the processing of object can not be terminated occurs.

[0004]

In a case that such a multi-occurrence deadlock occurs, such a method has been employed conventionally that the processing for one of the jobs is subjected to abnormal termination and the processing for the other job is continued.

(19)日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11)特許番号

特許第3005397号
(P3005397)

(45)発行日 平成12年1月31日(2000.1.31)

(24)登録日 平成11年11月19日(1999.11.19)

(51)Int.Cl.⁷

G 0 6 F 9/46
11/30

識別記号

3 4 0
3 0 5

F I

G 0 6 F 9/46
11/30

3 4 0 G
3 0 5 G

請求項の数4(全 4 頁)

(21)出願番号 特願平5-220764
(22)出願日 平成5年9月6日(1993.9.6)
(85)公開番号 特開平7-73053
(43)公開日 平成7年3月17日(1995.3.17)
審査請求日 平成9年12月16日(1997.12.16)

(73)特許権者 390001395
関西日本電気ソフトウェア株式会社
大阪府大阪市中央区城見1丁目4番24号
(72)発明者 中川 昇
和歌山県和歌山市八番丁11 和歌山日本
電気ソフトウェア株式会社内
(74)代理人 100082935
弁理士 京本 直樹

審査官 久保 光宏

(56)参考文献 特開 昭62-3367 (J P, A)
特開 平2-43641 (J P, A)
特開 平4-145538 (J P, A)
特開 平6-67935 (J P, A)

最終頁に続く

(54)【発明の名称】 デッドロック多発自動回避方式

1

(57)【特許請求の範囲】

【請求項1】 複数のジョブが共有の資源を排他的に使用して並列に処理を行う情報処理システムにおいて、各ジョブごとに実行中の処理区間で発生したデッドロックの発生回数を保持しておくデッドロック発生回数記憶テーブルと、デッドロックが発生したときに処理区間の開始点まで戻されて再実行を指示されたジョブに対応する前記デッドロック発生回数記憶テーブルの値に1を加えて更新し更新後の値があらかじめ定められた制限値に達した場合には当該ジョブを一時停止させるジョブ一時停止手段と、前記ジョブ一時停止手段により停止させられたジョブの一時停止時間を監視し規定時間が経過すると終了通知を出力する停止時間監視手段と、前記終了通知を受けて当該ジョブの一時停止を解除して前記デッドロック発生回数記憶テーブルの該当ジョブのデッドロ

2

ック発生回数を初期状態に戻すジョブ停止解除手段と、各ジョブ処理区間の処理終了ごとに前記デッドロック発生回数記憶テーブルの該当ジョブのデッドロック発生回数を初期状態に戻すジョブ処理区間終了処理手段とを備えたことを特徴とするデッドロック多発自動回避方式。

【請求項2】 前記ジョブ一時停止手段がジョブを一時停止させるための制限値が、ジョブごとにデッドロック発生回数制限値テーブルに設定されていることを特徴とする請求項1記載のデッドロック多発自動回避方式。

【請求項3】 前記停止時間監視手段がジョブの一時停止を解除するまでの規定時間がジョブごとにジョブ停止設定時間テーブルに設定されていることを特徴とする請求項1又は請求項2記載のデッドロック多発自動回避方式。

【請求項4】 少なくとも前記デッドロック発生回数制

限值テーブルのジョブごとの制限値が各ジョブの起動時にジョブ制御文によって設定されることを特徴とする請求項2又は請求項3記載のデッドロック多発自動回避方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はデッドロック多発自動回避方式に関し、特にデータベースのような共有の資源を複数のジョブが同時に使用し、排他制御を受けながら並列的に処理を行う情報処理システムにおいて、1処理区

【0002】

【従来の技術】二つ以上のジョブが同一資源を排他的に使用しながら並行して処理を行うデータベースシステムのような情報処理システムにおいては、二つのジョブがそれぞれ相手の占有している資源を要求し合って相互に待ち状態となるデッドロック現象が発生する。これを回避するための方策として、デッドロックが発生すると、一般に後から資源要求を行ってデッドロック状態を発生させた側のデッドロック発生ジョブに対して、現在実行中の処理区間の既に実行済みの処理を取り消してその処理区間の開始点から再実行させることが行われている。この対策により当面のデッドロックは通常これを回避することができる。

【0003】上述したように、処理区間内のある時点で発生したデッドロックはデッドロック発生ジョブを開始点から再実行させることにより回避され、デッドロックの相手となったデッドロック対象ジョブは、デッドロック発生時点の処理から引き続き処理を継続できるが、処理区間が長くて複雑な場合には、同じ処理区間内の他の時点で再度デッドロック現象が発生し、今度は自分がデッドロック発生ジョブであれば処理を継続することができずに処理区間の開始点まで戻されることになる。従って、二つのジョブがそれぞれ1処理区間内で交互に原因となってデッドロックを繰り返し発生した場合には、双方のジョブは交互に処理区間開始点からの開始処理（例えばデータベース更新処理）とデッドロックによる更新取消処理とを繰り返し行うだけで共に目的の処理を終了できなくなる現象（多発性デッドロックと呼ぶ）が発生する。

【0004】このような多発性デッドロックが発生した場合、従来は、どちらか一方のジョブを異常終了させることにより、もう一方のジョブの処理を継続させる方法が用いられていた。

【0005】

【発明が解決しようとする課題】上述した従来の多発性デッドロックの回避方式では、デッドロックが1処理区間内で多発した場合に、どちらか一方のジョブを強制的に異常終了させて実行を中止させるため、双方のジョブ

の処理を継続させることはできず、異常終了したジョブを実行させるためにはオペレータの介入が必要であった。

【0006】本発明の目的は、1処理区間内でデッドロックが多発した場合に、ジョブを異常終了させずに双方のジョブをオペレータの介入なく継続処理できるデッドロック多発自動回避方式を提供することにある。

【0007】

【課題を解決するための手段】請求項1のデッドロック多発自動回避方式は、複数のジョブが共有の資源を排他的に使用して並列に処理を行う情報処理システムにおいて、各ジョブごとに実行中の処理区間内で発生したデッドロックの発生回数を保持しておくデッドロック発生回数記憶テーブルと、デッドロックが発生したときに処理区間の開始点まで戻されて再実行を指示されたジョブに対応する前記デッドロック発生回数記憶テーブルの値に1を加えて更新し更新後の値があらかじめ定められた制限値に達した場合には当該ジョブを一時停止させるジョブ一時停止手段と、前記ジョブ一時停止手段により停止させられたジョブの一時停止時間を監視し規定時間が経過すると終了通知を出力する停止時間監視手段と、前記終了通知を受けて該当ジョブの一時停止を解除して前記デッドロック発生回数記憶テーブルの該当ジョブのデッドロック発生回数を初期状態に戻すジョブ停止解除手段と、各ジョブ処理区間の処理終了ごとに前記デッドロック発生回数記憶テーブルの該当ジョブのデッドロック発生回数を初期状態に戻すジョブ処理区間終了処理手段とを備えて構成されている。

【0008】請求項2のデッドロック多発自動回避方式は、請求項1記載のデッドロック多発自動回避方式において、前記ジョブ一時停止手段がジョブを一時停止させるための制限値が、ジョブごとにデッドロック発生回数制限値テーブルに設定されていることを特徴としている。

【0009】請求項3のデッドロック多発自動回避方式は、請求項1又は請求項2に記載したデッドロック多発自動回避方式において、前記停止時間監視手段がジョブの一時停止を解除するまでの規定時間が、ジョブごとにジョブ停止設定時間テーブルに設定されていることを特徴としている。

【0010】請求項4のデッドロック多発自動回避方式は、請求項2又は請求項3に記載したデッドロック多発自動回避方式において、少なくとも前記デッドロック発生回数制限値テーブルのジョブごとの制限値が、各ジョブの起動時にジョブ制御文により設定されることを特徴としている。

【0011】

【実施例】次に、本発明の実施例について図面を参照して説明する。

【0012】図1は本発明の一実施例の構成を示すプロ

10

20

30

40

50

ック図である。

【0013】本実施例のデッドロック多発自動回避方式は、図1に示すように、二つのジョブ、デッドロック発生ジョブ8とデッドロック対象ジョブ9とがデータベース資源10を排他的に使用して更新を行う情報処理システムにおいて、各ジョブの実行中の処理区間内で発生したデッドロックの発生回数を保持しておくデッドロック発生回数記憶テーブル5と、デッドロックが発生したときに処理区間の開始点まで戻され再実行を指示されたジョブに対応するデッドロック発生回数記憶テーブル5の発生回数に1を加算して更新し、デッドロック発生回数制限値テーブル6に設定された制限値に達した場合には当該ジョブを一時停止させるジョブ一時停止手段1と、停止させられたジョブの一時停止時間を監視しジョブ停止設定時間テーブル7に設定された規定時間が経過すると終了通知を出力する停止時間監視手段2と、終了通知を受け該当ジョブの一時停止を解除すると共にデッドロック発生回数記憶テーブル5の該当ジョブのデッドロック発生回数を初期状態に戻すジョブ停止解除手段3と、各ジョブ処理区間の処理終了ごとにデッドロック発生回数記憶テーブル5の該当ジョブのデッドロック発生回数を初期化するジョブ処理区間終了処理手段4とを備えて構成されている。

【0014】システムで実行中の二つのジョブ、デッドロック発生ジョブ8とデッドロック対象ジョブ9とが共にデータベース資源10の更新処理を行っているとき、デッドロック対象ジョブ9を相手にデッドロック発生ジョブ8にデッドロックが発生すると、デッドロック発生ジョブ8はシステムにより処理区間の開始点まで処理を取り消され、ジョブ一時停止手段1にデッドロック発生を通知する。

【0015】ジョブ一時停止手段1は、デッドロック発生回数記憶テーブル5内にあるデッドロック発生ジョブ8のデッドロック発生回数に1を加えて更新し、その更新した発生回数とデッドロック発生回数制限値テーブル6中のデッドロック発生ジョブ8の制限回数とを比較する。発生回数が制限回数に達していない場合は、そのままデッドロック発生ジョブ8に処理区間の開始点からデータベース資源10の更新処理を再実行させる。発生回数が制限回数に達していた場合には、ジョブ一時停止手段1はデッドロック発生ジョブ8の処理を一時停止させ、停止時間監視手段2にデッドロック発生ジョブ8を停止させたことを通知する。

【0016】停止時間監視手段2は、デッドロック発生ジョブ8の停止時間を監視し、停止時間がジョブ停止設定時間テーブル7に設定されているデッドロック発生ジョブ8の規定時間を超えるとジョブ停止解除手段3に通知する。

【0017】ジョブ停止解除手段3は、デッドロック発生ジョブ8の一時停止を解除すると共に、デッドロック

発生回数記憶テーブル5内にあるデッドロック発生ジョブ8のデッドロック発生回数を初期状態(0)に変更する。なお、デッドロック発生回数記憶テーブル5内の各ジョブのデッドロック発生回数は、各ジョブの処理区間が終了することにジョブ処理区間終了処理手段4によりそれぞれ初期状態に変更される。

【0018】以上、デッドロック発生ジョブ8を中心に説明したが、デッドロック対象ジョブ9についても同様の処理が行われる。このように、1処理区間内でデッドロックが多発し発生回数が規定値に達した場合、ジョブの処理を一時的に停止して処理区間開始のタイミングを相手ジョブとずらすことにより、二つのジョブが交互に同一処理区間の処理を途中まで繰り返す多発性デッドロックをオペレータの介入なしに回避し、双方のジョブを継続させることが可能となる。

【0019】上述の実施例では、デッドロック発生回数制限値テーブル6と停止時間がジョブ停止設定時間テーブル7とを備え、それぞれに各ジョブごとの制限値と規定時間とが設定されており、ジョブ一時停止手段1及び停止時間監視手段2は各ジョブに対応する制限値および規定時間を参照して一時停止および解除通知を行うものとしたが、各ジョブごとに制限値および規定時間を設定するのではなく、テーブルを設けることなくすべてのジョブに対して一律に同じ値を適用しても同様な効果を期待することができる。

【0020】上述の実施例のごとくテーブルを設けてジョブごとに異なる値を設定できるようにすると、処理区間の長さやジョブの重要度に応じて設定値を変えることができるため、より効率的な運用が可能となる利点がある。テーブルには特定のジョブに対する設定値のみを登録しておき、登録されていないジョブに対しては一律の値(例えば、小さい制限値と長めの規定時間)を適用するようにしてもよい。なお、テーブルの設定値は別途設定しておいてもよいが、ジョブの起動時にジョブ制御文により登録するようにすると、余分な設定工数やメモリ領域を必要とせず効果的な運用ができる。

【0021】

【発明の効果】以上説明したように、本発明のデッドロック多発自動回避方式は、データベース更新処理のように同一資源を複数のジョブが同時に排他的に使用している場合に、双方のジョブが交互に同一処理区間でデッドロックを繰り返しジョブが終了しなくなる多発性デッドロック現象の発生を、オペレータが介入することなく自動的に回避することができ、双方のジョブを異常終了させずに継続処理できる効果がある。

【図面の簡単な説明】

【図1】本発明の一実施例の構成を示すブロック図である。

【符号の説明】

1 ジョブ一時停止手段

